YF 小型推荐系统

目录

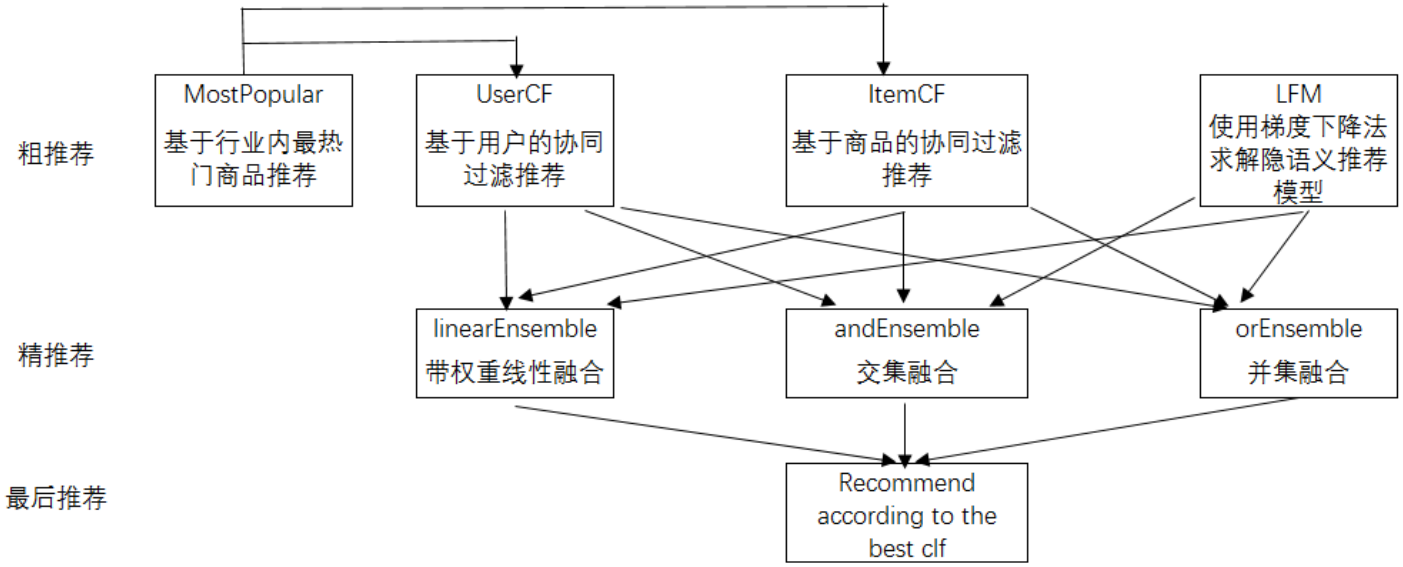
一. 开发背景

YF 公司，是一家做食品添加剂零售与批发的公司，其母公司在台湾，集生、产、销于一体，在食品添加剂行业内属于排名较前的公司。当时在 YF 公司主要做销售助理的工作，所以对 YF 公司的业务比较熟悉。当时，我就发现我们公司拥有了一些比较优质的客户，每次下订单数量比较大，购买关系稳定，但是他们购买的添加剂种类不是很多，常常是固定的那几支添加剂。当时就想，为什么不推荐一些公司的爆款产品给他们呢，也许推荐了以后他们就会买呢。然后，我就通过 AS400 系统手动查询，并罗列了适合几家客户的推荐清单，并且建议给我们研发经理，当时我们公司的研发经理是由台湾外派过来的，同时担任了研发和销售的工作，然后受到了研发经理的好评。结果，我们研发经理就带领研发团队，开发了好多含有推荐的添加剂的食品，然后对这些客户，甚至更多地其他客户，进行推荐。据说，最终收获非常不错，甚至还拿到几家公司的战略合作关系，即在推荐食品添加剂的同时为他们开发新的小食品，当时作为试吃的我们，每天都能吃到我们研发团队开发出来的小食品，或面包、或饼干、或糕点、或布丁、或饮料、或咖啡、或香肠、或肉串等等，至今难忘。

所以，现在想利用自己的知识，开发一款适合 YF 公司的小型推荐系统。它不仅可以大大减少人工手动查询什么添加剂适合推荐什么公司的繁琐与劳动力，同时能够大大提升推荐的精度。YF 公司一直处于很难招聘到拥有专业的食品添加剂知识的销售人员，那么有了推荐系统之后，甚至可以不用再去招聘销售人员，可以将招聘销售人员的成本转到招聘研发人员上，从而使研发团队与推荐系统的作完美配合即可。

二. 系统设计

1. 系统模型设计图



2. 推荐算法说明

	第一层：粗推荐		
算法	UserCF 基于用户的协同过滤	ItemCF 基于商品的协同过滤	LFM 基于矩阵分解的推荐
类型	基于统计的推荐	基于统计的推荐	基于机器学习模型
属性	偏群体化推荐	偏个体化推荐	让模型自学习后推荐
适用对象	较适用于定量、定类购买的大客户	较适用于询价多、购买量少的小客户	均适用
说明	建立在相似用户的基础上，偏向于推荐群体性喜欢的商品，对用户个性化兴趣不太明显时使用	建立在相似商品的基础上，可以针对询价而未购买的商品的相似商品进行推荐，如果结合价格粗略，则效果应当会更好	挖掘出肉眼看不到的用户与商品的联系，并使用了 SGD 进行优化求解
评估结果	流行度最高	覆盖率最高	由于虚拟数据比较少，所以得到的结果不甚理想
用户冷启动策略	对于新用户，使用了 MostPopular 推荐策略，即推荐行业内最热门的商品	对于新用户，使用了 MostPopular 推荐策略，即推荐行业内最热门的商品	本身就带有隐类在里面，所以没有使用 MostPopular 来进行用户冷启动处理

	第二层：精推荐		
融合方式	线性融合（带权重）	交集融合	并集融合
融合机制	根据离线实验结果，给予相应的权重，同时将这个权重值赋值给了推荐列表中的 item，然后再使用统计算法为 item 进行排序，取 topN	取三个算法中所有推荐的商品，然后再使用他们出现在推荐列表中的热度进行排序，取 topN	取三个算法中都被推荐到的商品，如果数量比较多，则再按照他们的热度进行排序，取 topN；如果数量比较少，则按照被推荐次数进行补位；
评估结果	覆盖率指标，有非常大提升	流行度指标，有非常大提升，两者融合方式得到的结果非常相近	
融合优点	较大提高推荐覆盖率	较大提高推荐流行度	

3. 系统整体模块展示

系统 github 代码：https://github.com/Tntade/Recommendation

4. 离线实验结果展示与说明

（1） 在评估模型指标的时候，采用了 bagging 思想，每一次运行给予不同的 seed 来达到模拟随机重采样的效果，然后对评估指标取平均值；

（2） 实验结果如下图：

按照 Coverage 覆盖率排序：线性融合推荐，效果是最好的；

	alg	avg_coverage_on_30_exp	avg_popularity_on_30_exp	avg_times_on_30_exp	avg_precision_on_30_exp	avg_recall_on_30_exp
3	linearEnsemble	61.000000	2.755320	0.292850	0.0	0.0
0	ItemCF	59.611111	2.627222	0.019434	0.0	0.0
2	LFM	51.611111	1.578611	0.266315	0.0	0.0
1	UserCF	46.277778	3.519600	0.011601	0.0	0.0
5	orEnsemble	36.111111	3.567574	0.299184	0.0	0.0
4	andEnsemble	34.500000	3.440891	0.290550	0.0	0.0

按照 Popularity 流行度排序：交集融合和并集融合推荐，效果是最好的，且两者效果相近；

	alg	avg_popularity_on_30_exp	avg_coverage_on_30_exp	avg_times_on_30_exp	avg_precision_on_30_exp	avg_recall_on_30_exp
5	orEnsemble	3.567574	36.111111	0.299184	0.0	0.0
1	UserCF	3.519600	46.277778	0.011601	0.0	0.0
4	andEnsemble	3.440891	34.500000	0.290550	0.0	0.0
3	linearEnsemble	2.755320	61.000000	0.292850	0.0	0.0
0	ItemCF	2.627222	59.611111	0.019434	0.0	0.0
2	LFM	1.578611	51.611111	0.266315	0.0	0.0

说明： Precision 与 Recall 均为 0 的主要原因，由以下几个方面结合导致：

a. 样本数量太少，只虚拟了 155 个样本，其中 109 个为产生购买行为的样本，其余为询价样本；

b. 所有样本均为随机虚构，但是，YF 公司的产品与用户所在行业是有一定联系的，比如，主营业务为烘焙的小企业不太可能会去购买饮料类的添加剂，而虚构样本并没有考虑这样的关系；

c. 109 个购买样本中，共涉及 28 个用户，平均每个用户购买产品数量为 3.9 个，precision 和 recall 都是针对验证集来说的，验证集随机采样比例为 20%，那么验证集中的样本用户购买的产品基本保持在 0 个或者 1 个，而我们推荐给用户的有 5 个（第一层中三个算法，topN 推荐为 5 只产品）或 3 个（第二层中三个融合算法，topN 推荐为 3 只产品），那么势必导致 precision 和 recall 等于 0 或趋近于 0。

三．系统优化与思考

1．系统功能优化与思考

（1）相似度的度量方式的改进：

度量方式的改进，如果可以使用余弦相似度/改进的余弦相似度/Jaccard 相似度/Pearson 相关系数等，或，取几者的平均值，则使得刻画地更加精确。

（2）通过机器学习算法，对用户进行分群：

根据 YF 公司用户的行为属性，将用户进行分群，然后对于不同的用户群采用不同的推荐算法。比如，对于 YF 优质的大客户群体，他们的行为属性是采购产品的种类少，但是采购数量大，那么 UserCF 推荐算法更加适合他们；而对于 YF 大量的小企业客户群体，他们的行为属性是询价很多、实际下单很少、对于已经购买过的产品，其采购数量非常小，且非常关注产品的价格，那么 ItemCF 推荐算法，结合一定的价格策略，就非常适合他们。

（3）融入关联规则算法：

利用关联规则算法，看看是否有关联规则产品对存在。如果有，则一旦推荐列表中出现了前件项，那么系统自动追加后件项到该用户的推荐列表中；

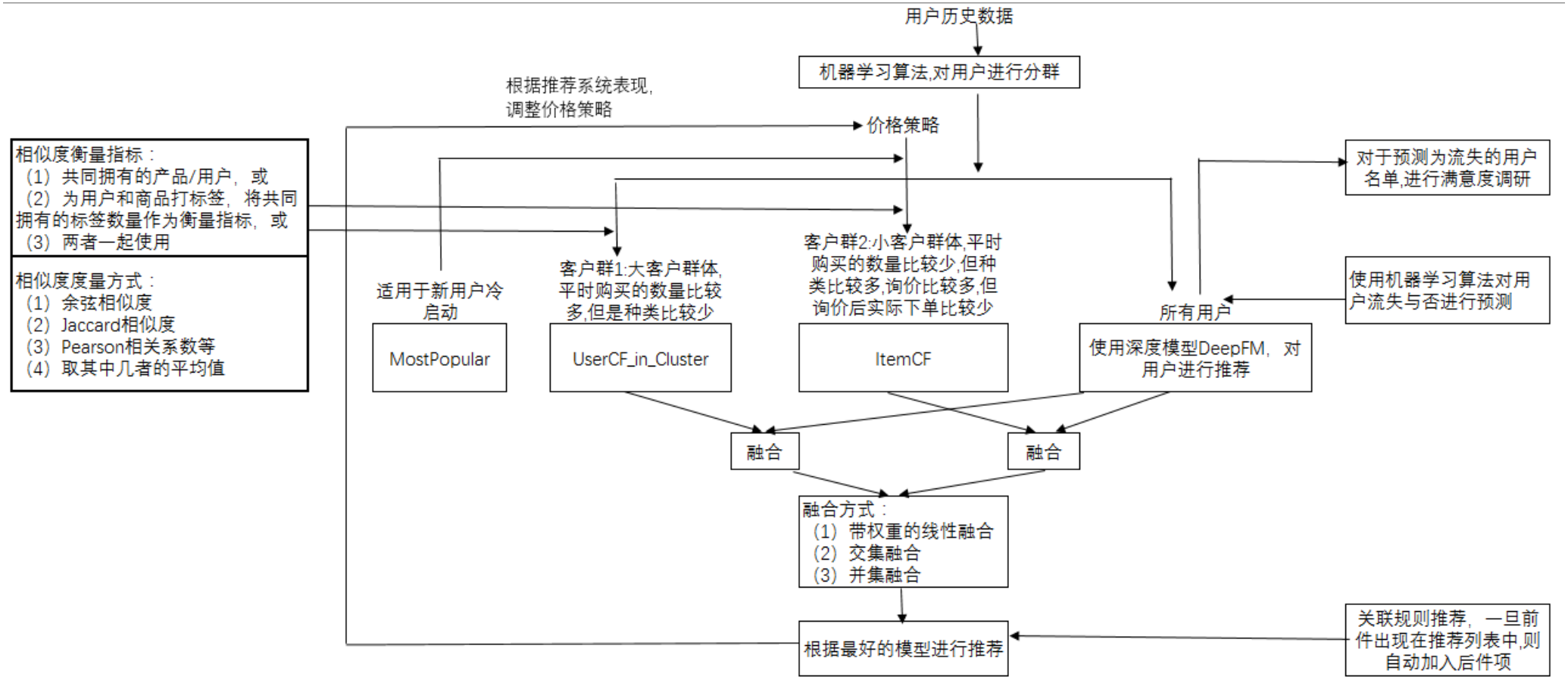
（4）融入用户流失判断机制+用户满意度调研

如果我们在系统后台就融入用户流失与否的预测功能，则根据预测提前得到预警，对于预测为流失的用户名单，及时进行客户电话回访及满意度调研，然后去改善、优化推荐策略；甚至自动启动推荐系统弹窗通知功能，直接推荐用户最想要的商品、并且以鲜美的价格吸引他，从而达到留住用户的效果。

（5）价格策略：

对于部分小企业用户，平时询价比较多，但实际询价后下单的比较少，较大原因是因为价格问题，那么对这些用户，对他们历史购买数据进行了解，确认基于公司正常报价的上下浮动范围；比如，C2018001 属于 C 级别客户，按照公司报价政策，对于 C 级别客户，报价策略为在产品成本价基础上浮 25%，可往往这样的策略报价，C2018001 基本不会购买，那么可以查看其历史数据，确定他一般可接受的下浮范围，比如在我们正常报价下浮 10%，他们一般会比较容易接受，那么就推荐这个价格范围的相似产品，如果没有，则推荐相似产品的同时给予一定的价格优惠。

2．根据以上思考，对系统模型进行了重设计：



四．自荐信

敬爱的人事及主管，

　　您好，非常感谢您看完我的 YF 小型推荐系统的方案。我知道这方案中肯定有许多的不足，但是，依然希望您能够给我一次进一步了解的机会。正如您所看到的，在过往的 2 年中，我利用了自己的业余时间，从零开始学习，从不懂 python 的小白，到现在可以快乐地写出自己想要的程序；从没参加过数据挖掘比赛，到多次在不同类型数据挖掘比赛中拿到个人首战 top10%的成绩；从不懂算法为何物，到现在可以历数机器学习经典算法流程，甚至还能搭建基础 CNN、RNN、LSTM 深度模型，并且在看到牛逼的算法、精辟的代码、创造性的模型设计时，能发出深深的感叹与崇拜，同时也期望并致力于将来的自己也能成为这样的人才，从而开发出优质的推荐系统，让用户得到更好的购买体验。希望您能给我一次面试的机会，给我们双方有一个更加深入了解的机会，谢谢。

陆琴